Distributed memory parallel implementation of energies and gradients for second-order Møller–Plesset perturbation theory with the resolution-of-the-identity approximation

Christof Hättig, Arnim Hellweg and Andreas Köhn†

Received 31st October 2005, Accepted 9th January 2006
First published as an Advance Article on the web 31st January 2006
DOI: 10.1039/b515355g

We present a parallel implementation of second-order Møller–Plesset perturbation theory with the resolution-of-the-identity approximation (RI-MP2). The implementation is based on a recent improved sequential implementation of RI-MP2 within the Turbomole program package and employs the message passing interface (MPI) standard for communication between distributed memory nodes. The parallel implementation extends the applicability of canonical MP2 to considerably larger systems. Examples are presented for full geometry optimizations with up to 60 atoms and 3300 basis functions and MP2 energy calculations with more than 200 atoms and 7000 basis functions.

Introduction

During the last decade the resolution-of-the-identity (RI) approximation[1–3] has developed as an important and widely used route to reduce the computational costs not only for density functional theory[4,5] but also for second-order Møller–Plesset perturbation theory (MP2) and related methods. While early applications were hampered by problems with the basis sets in which the resolution of the identity is done, these difficulties have in the meantime been overcome with the introduction of optimized special purpose auxiliary basis sets.[6] Today, accurate and efficient auxiliary basis sets are available for a variety of orbital basis sets[6–8] and allow to carry out RI-MP2 calculations with computational costs reduced by about an order of magnitude compared to conventional MP2 calculations while the additional error due to the RI approximation is kept small compared to the orbital basis set error. Several implementations of RI-MP2—recently also denoted as density fitting (DF) MP2—have been reported; among them a recent implementation of a local correlation variant[9,10] and explicitly correlated versions.[11,12]

The first parallel implementation of RI-MP2 energies for distributed memory architectures was reported a few years ago by Bernholdt *et al.*[13,14] based on the Global Array toolkit. We report here the parallel implementation of RI-MP2 gradients for distributed memory systems. Our work builds upon the algorithms reported originally by Weigend and Häser[15] for the sequential implementation of RI-MP2 gradients. In recent years, these algorithms have been revised to further improve memory management, integral evaluation and symmetry treatment. The efficiency of the sequential code, available now in the ricc2 module of the Turbomole package,[16,17] has recently been demonstrated in a comparison with modern conventional integral-direct MP2 implementations.[18]

As the simplest and computationally cheapest wave function based electron correlation method, second-order Møller–Plesset perturbation theory is, besides Hartree–Fock and density functional theory, one of the most widely applied electronic structure methods. While for many standard applications local MP2 (LMP2) provides a valuable and cheaper alternative to canonical MP2 (*i.e.* MP2 based on canonical Hartree–Fock orbitals), the improvement of the implementation of canonical MP2 is still an important task since LMP2 is not applicable in every situation. The local correlation approach does not yet give satisfactory results for, in particular, response calculations of *e.g.* excitation energies.[19,20] There are, however, several important MP2-related second-order methods for excited states derived from response or propagator theory as *e.g.* the perturbative doubles correction for configuration interaction singles (CIS(D)),[21] the approximate coupled-cluster singles-and-doubles model[22] (CC2), the algebraic diagrammatic construction through second order[23,24] (ADC(2)), or the second-order polarization propagator approach[25,26] (SOPPA). The possibility of a future extension to these excited state methods has been an important aspect of the present work, as our starting point was a sequential program—the ricc2 module of Turbomole—which was originally developed for calculations of excited states.

The paper is organized as follows: In the next section the basic equations for analytic gradients of the RI-MP2 energy are briefly reviewed and in section III the distributed memory parallel implementation is described. The performance of the implementation for RI-MP2 energy and gradient calculations is analyzed in section IV for six test examples, and the first applications which demonstrate how the parallel implementation extends the applicability of MP2 to larger systems are presented in section V. A short summary and some conclusions are given in section VI.

Forschungszentrum Karlsruhe, Institute of Nanotechnology, P. O. Box 3640, D-76021 Karlsruhe, Germany
† Present adress: Department of Chemistry, University of Århus, DK-8000 Århus C, Denmark

This journal is © the Owner Societies 2006

II. Analytic gradients for RI-MP2

In second-order Møller–Plesset perturbation theory the correlation energy for a closed shell system is calculated by

$$E_{\mathrm{MP2}} = \sum_{ia \geq jb} \frac{(ia|jb)[2(ia|jb) - (ib|ja)]}{\varepsilon_i - \varepsilon_a + \varepsilon_j - \varepsilon_b}, \qquad (1)$$

where $(ia|jb)$ are two-electron repulsion integrals (ERIs) in Mulliken (or charge cloud) notation and ε_p are canonical self-consistent (SCF) orbital energies. We use the convention that the indices i, j,... denote active occupied and a, b,... active unoccupied (virtual) orbitals. The indices p, q, will be used for general molecular orbitals (MO) and μ, ν, κ, λ for atomic orbitals (AO).

In canonical orbital based MP2 energy calculations the time-determining step is the transformation of 4-index ERIs—evaluated in the AO basis—to the MO basis.

$$(ia|jb) = \sum_{\nu} C_{\nu a} \sum_{\lambda} C_{b\lambda} \sum_{\mu} C_{\mu i} \sum_{\kappa} C_{j\kappa} (\mu\nu|\kappa\lambda) \qquad (2)$$

The operation count of the 4-index transformation is nominally $1/2nN^4 + 3/2n^2\ N^3$, where n is the number occupied orbitals and N the total number basis functions (assuming that $n \ll N$ and thus the number of virtual orbitals $V \approx N$); furthermore we use in the following $\mathcal{N}$ to indicate the scaling of computational costs with the system size. With integral prescreening based on the Schwartz inequality[27] and other estimates, the computational costs of, in particular, the first partial transformation ($\mathcal{O}(nN^4)$) can be reduced significantly.[28,29] Additional severe bottlenecks for large scale applications are the huge storage requirements of $\mathcal{O}(n^2N^2)$ for such a 4-index transformation and related I/O and CPU demands for intermediate sorting steps needed in multipass algorithms for large cases.[28,29]

In RI-MP2 the evaluation of the 4-index ERIs in the MO basis $(ia|jb)$ is accelerated and the above mentioned storage bottleneck bypassed by employing the RI approximation for ERIs[1–3]

$$(\mu\nu|\kappa\lambda) \approx \sum_{PQ} (\mu\nu|P)[V^{-1}]_{PQ}(Q|\kappa\lambda) \qquad (3)$$

with the Coulomb metric

$$V_{PQ} = (P|Q). \qquad (4)$$

In the latter equations $(\mu\nu|P)$ and $(P|Q)$ are 3- and 2-index ERIs and P, Q are functions of an optimized auxiliary basis set.[6–8] Within the RI approximation, 4-index ERIs in the MO basis $(ia|jb)$ can be evaluated with high efficiency from 3-index intermediates by a single matrix–matrix multiplication

$$(ia|jb) \approx (ia|jb)_{\mathrm{RI}} = \sum_{Q} B^i_{Qa} B^j_{Qb} \qquad (5)$$

with an operation count $\approx 1/2n^2N^2N_{\mathrm{x}}$, where N_{x} is the size of the auxiliary basis. The intermediates B^i_{Qa}, defined as

$$B^i_{Qa} = \sum_{P} [V^{-1/2}]_{QP} \sum_{\mu} C_{\mu a} \sum_{\nu} C_{\nu i} (\nu\mu|P), \qquad (6)$$

can be calculated with $\mathcal{O}(nN^3)$ costs and require only $\mathcal{O}(nN^2)$ disk space for storage—in contrast to the $\mathcal{O}(n^2N^2)$ storage demands for the 4-index transformation in eqn (2). If the auxiliary basis sets are carefully optimized the number of auxiliary basis functions N_{x} can usually be kept within $1.5N \leq N_{\mathrm{x}} \leq 4N$ (depending on the one-electron orbital basis and the atom) while the error introduced by the RI approximation in the MP2 correlation energy is kept to less than 1% of the respective orbital basis set errors.[6,7,18] The reduction in operation count obtained with the RI approximation depends on the size of the orbital and auxiliary basis sets; it scales roughly with N^2/nN_{x}. In triple-ζ basis sets—as often used for MP2 calculations—RI-MP2 is typically about an order of magnitude faster than a conventional MP2 code, but with larger orbital basis sets or if the orbital basis contains diffuse functions that hinder the integral prescreening, much larger reductions are often found.

If the total energy is defined as the sum of the RI-MP2 correlation energy and a conventional (*i.e.* non-RI) SCF energy, the analytic gradients can be expressed as[15,30]

$$\begin{aligned} \frac{dE_{\mathrm{SCF+MP2}}}{dx} = & \sum_{\mu\nu} D^{\mathrm{eff,ao}}_{\mu\nu} h^{[x]}_{\mu\nu} + \frac{1}{2} \sum_{\mu\nu\kappa\lambda} d^{\mathrm{sep,ao}}_{\mu\nu\kappa\lambda} (\mu\nu|\kappa\lambda)^{[x]} \\ & - \sum_{\mu\nu} F^{\mathrm{eff,ao}}_{\mu\nu} S^{[x]}_{\mu\nu} + \sum_{\mu\nu P} \Delta^{\mathrm{ao}}_{P,\mu\nu} (\mu\nu|P)^{[x]} \\ & - \sum_{PQ} \gamma_{PQ} (P|Q)^{[x]}. \end{aligned} \qquad (7)$$

In the above equation $h^{[x]}_{\mu\nu}$ and $S^{[x]}_{\mu\nu}$ are the derivatives of the one-electron Hamiltonian and overlap AO integrals with respect to a coordinate x, respectively. The terms $(\mu\nu|\kappa\lambda)^{[x]}$, $(\mu\nu|P)^{[x]}$, and $(P|Q)^{[x]}$ are the derivatives of the 4-, 3-, and 2-index ERIs in the AO basis. The 3- and 2-index two-electron densities are defined as

$$\Delta^{\mathrm{ao}}_{P,\mu\nu} = \sum_{i} \Gamma_{Pi\nu} C_{\mu i}, \qquad (8)$$

$$\gamma_{PQ} = \sum_{R} [V^{-1/2}]_{RP} \sum_{ia} B^i_{Ra} \sum_{S} [V^{-1/2}]_{SQ} Y^i_{Sa}, \qquad (9)$$

with

$$Y^i_{Qa} = \sum_{jb} \frac{2(ia|jb) - (ib|ja)}{\varepsilon_a - \varepsilon_i + \varepsilon_b - \varepsilon_j} B^j_{Qa}, \qquad (10)$$

and

$$\Gamma_{Pi\nu} = \sum_{a} C_{\nu a} \sum_{Q} [V^{-1/2}]_{PQ} Y^i_{Qa}. \qquad (11)$$

(The intermediate $V^{-1/2}$ is replaced in the implementation by the Cholesky decomposition of V^{-1}, which is computationally more efficient.) The separable part of the two-electron density is obtained as usual by

$$d^{\mathrm{sep,ao}}_{\mu\nu\kappa\lambda} = \left(1 - \frac{1}{2}\hat{P}_{\nu\lambda}\right)\left(D^{\mathrm{eff,ao}}_{\mu\nu} - \frac{1}{2} D^{\mathrm{SCF,ao}}_{\mu\nu}\right) D^{\mathrm{SCF,ao}}_{\kappa\lambda}, \qquad (12)$$

where the permutation operator is defined by $\hat{P}_{\mu\lambda} f_{\mu\lambda} = f_{\lambda\mu}$ and $D^{\mathrm{SCF,ao}}$ and $D^{\mathrm{eff,ao}}$ are the SCF and the effective or orbital-relaxed MP2 one-electron densities in the atomic orbital basis.

This journal is © the Owner Societies 2006

The latter density is obtained as the sum

$$D_{pq}^{\mathrm{eff}} = D_{pq}^{\mathrm{SCF}} + D_{pq}^{\mathrm{MP2}} + \frac{1}{2}(\bar{\kappa}_{pq} + \bar{\kappa}_{qp}) \qquad (13)$$

of the SCF density, the unrelaxed MP2 correlation contribution

$$D_{ij}^{\mathrm{MP2}} = -\sum_{abk} (2t_{ab}^{jk} - t_{ba}^{jk}) t_{ab}^{ik}, \qquad (14)$$

$$D_{ab}^{\mathrm{MP2}} = +\sum_{ijc} (2t_{ac}^{ij} - t_{ac}^{ij}) t_{bc}^{ij}, \qquad (15)$$

(all other elements of D^{MP2} are zero), and a correction from the orbital-relaxation Lagrange multipliers $\bar{\kappa}$ which are obtained as solutions to the so-called Z-vector equations

$$\sum_{AIBJ} \bar{\kappa}_{AI}(A_{AIBJ} + \delta_{AB}\varepsilon_A - \delta_{IJ}\varepsilon_I) = -\eta_{BJ}^{\kappa}. \qquad (16)$$

Indices I, J and A, B denote here the general (*i.e.* active and frozen) occupied and virtual indices, respectively, and the coupled perturbed Hartree–Fock (CPHF) matrix A_{pqrs} is defined as $A_{pqrs} = 4(pq|rs) - (pr|sq) - (ps|rq)$ with conventional 4-index ERIs. The right-hand side vector η^{κ} and the effective Fock matrix F^{eff} (also known as the energy weighted density matrix) can be calculated from the above one-electron densities and 3-index intermediates by

$$\eta_{AI}^{\kappa} = L_{AI}^{''} - L_{IA} - \frac{1}{2}\sum_{pq} D_{pq}^{\mathrm{MP2}} A_{pqAI} \qquad (17)$$

$$F_{\alpha\beta}^{\mathrm{eff,ao}} = \sum_{pq} C_{\alpha p} C_{\beta q} \varepsilon_p D_{pq}^{\mathrm{eff}} + \sum_{Iq} C_{\alpha I} C_{\beta q} \sum_{rs} (D_{rs}^{\mathrm{MP2}} + \bar{\kappa}_{rs}) A_{rsIq} + \frac{1}{2}\sum_{pq} C_{\alpha p} C_{\beta q}\left\{L_{pq}^{''} + L_{pq}\right\} \qquad (18)$$

with

$$L_{iq} = \sum_{Qa} Y_{Qa}^{i} B_{Qa}^{q} = \sum_{\mu} C_{\mu q} \sum_{P\nu} (P|\nu\mu) \Gamma_{Piv} \qquad (19)$$

and

$$L_{aq}^{''} = \sum_{Qi} Y_{Qa}^{i} B_{Qq}^{i} \qquad (20)$$

and all other elements of L_{pq} and $L_{pq}^{''}$ defined as zero. The sequential implementation of the above equations has been described by Weigend and Häser in ref. 31. Generalizations to RI-CC2 for ground and excited states and RI-CIS(D_{∞}) and RI-ADC(2) have been derived in ref. 30, 32 and 33. (Note that in the latter references the notation for L and L'' has been changed in the course of the generalizations to $\Omega^{(b)}$ and $H^{(b)}$, respectively.)

III. Parallel implementation for distributed memory architectures

From the equations reviewed in section II it follows that the time-determining steps in RI-MP2 gradient calculations for large molecules (*i.e.* $n^2 \gg N$) are the computation of the approximated ERIs $(ia|jb)_{\mathrm{RI}}$ and their contraction to the intermediates Y_{Qa}^{i}, D_{ij}^{MP2}, and D_{ab}^{MP2}. These are the steps for which the computational costs increase as $\mathcal{O}(n^2N^3)$. As described in ref. 30, 31, 34 and 35, Y_{Qa}^{i} and D_{ab}^{MP2} can efficiently be calculated in a loop over two occupied indices with $\mathcal{O}(N^2)$ memory demands. For the occupied/occupied block of the unrelaxed density matrix D_{ij}^{MP2} this is only possible in a canonical reformulation[31] of the definition in eqn (14). For $|\varepsilon_i - \varepsilon_j| \geq t_{\mathrm{can}}$, where t_{can} is a threshold chosen large enough to ensure numerical stability, the density matrix elements D_{ij}^{MP2} can be calculated as

$$D_{ij}^{\mathrm{MP2}} = -\sum_{abk} (2t_{ab}^{jk} - t_{ba}^{jk}) t_{ab}^{ik} = 2(L_{ij} - L_{ji})/(\varepsilon_j - \varepsilon_i). \qquad (21)$$

Only matrix elements between nearly degenerate orbitals are calculated using the expression in eqn (14). These are usually only a small fraction of the total number of matrix elements and can be evaluated together with Y_{Qa}^{i} and D_{ab}^{MP2} in a loop over two occupied orbital indices without sacrificing the high efficiency of RI-MP2 or its low storage requirements.

We can thus conclude that the time-determining steps of RI-MP2 are most efficiently parallelized over pairs of occupied orbital indices since these are common to all $\mathcal{O}(n^2N^3)$ scaling steps. An alternative would be pairs of virtual orbital indices, but this would result in short loop lengths and diminished efficiency for medium sized molecules. A parallelization over auxiliary basis functions would require the communication of 4-index MO integrals and thus transfer rates which can only be reached with high performance networks. Such a solution would restrict the applicability of the program to high end supercomputer architectures. We, however, are seeking a solution that also performs well on low cost PC clusters with standard networks (*e.g.* Fast Ethernet or Gigabit) and accept that this results in an implementation that will not be suited for massively parallel systems.

A key problem for the parallelization of RI-MP2 is thus the distribution of pairs of occupied indices (ij) over distributed memory nodes such that

(a) the symmetry of $(ia|jb)$ with respect to permutation of $ia \leftrightarrow jb$ can still be exploited, and

(b) the demands on the individual computer nodes for accessing and/or storing the 3-index intermediates B_{Qa}^{i} and Y_{Qa}^{i} are as low as possible.

To achieve this, we partition the occupied orbitals into n_{CPU} batches $\mathscr{I}_m$ of (as much as possible) equal size, where n_{CPU} is the number of computer nodes. The pairs of batches $(\mathscr{I}_m, \mathscr{I}_{m'})$ with $m \leq m'$ can be ordered either on the upper triangle of a symmetric matrix or on block diagonal stripes as shown in Fig. 1. Now each computer node is assigned in a suitable way to one block out of each diagonal, such that each node only needs access to a minimal number of batches $\mathscr{I}_m$ of B_{Qa}^{i} and Y_{Qa}^{i} as shown, *e.g.*, for 13 nodes in Fig. 2. The minimal number of batches a node needs to access—in the following denoted as n_{blk}—increases approximately with $\sqrt{n_{\mathrm{CPU}}}$. The calculation of these 3-index ERIs B_{Qa}^{i} would require about $\mathcal{O}(N^3) + \mathcal{O}(nN^3) \times n_{blk}/n_{\mathrm{CPU}}$ floating point multiplications. Similar computational costs arise for the calculation of $\Delta_{P,\mu\nu}^{\mathrm{ao}}$ from Y_{Qa}^{i}. Thus, a conflict between minimization of the operation count and communication arises:

• If the 3-index intermediates are communicated between the nodes to avoid multiple integral evaluations, the

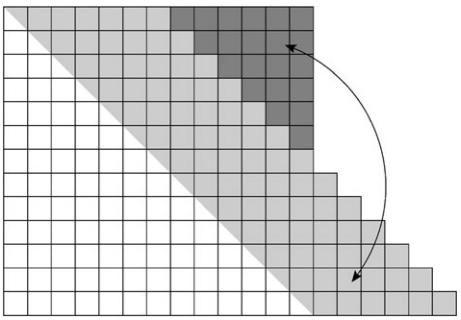


Fig. 1 Arrangement of the pairs of batches $m \leq m'$ with active occupied orbitals on the upper triangle of a symmetric matrix or on block diagonal stripes.

communication demands per node become relatively large ($\sim NN_x \times n/\sqrt{n_{CPU}}$).

• If the communication of 3-index intermediates is avoided by evaluating all integrals needed on each node, the operation count for the steps, which in RI-MP2 energy calculations are the next expensive ones after the $\mathcal{O}(n^2N^3)$ steps, only decreases with $\sqrt{n_{CPU}}$.

The first option requires a high bandwidth for communication while the second option can also be realized with a low bandwidth, but at the expense of a less efficient parallelization. For both ways a prerequisite for a satisfactory efficiency is that the total computational costs are dominated by those for the $\mathcal{O}(\mathcal{N}^5)$ steps such that the time needed for multiple calculations ($\mathcal{O}(\mathcal{N}^4)$) or communication ($\mathcal{O}(\mathcal{N}^3)$) of 3-index intermediates is a negligible fraction of the overall time of the RI-MP2 calculation. Both options have been realized in our parallel implementation of RI-MP2 and shall in the following be denoted as modes for ''slow communication'' and ''fast communication''. These considerations are obsolete for architectures with a fast global parallel file system which can be used for the storage of the 3-index intermediates and thereby makes an explicit communication of these intermediates unnecessary. For this case the same algorithms are used as in the fast communication mode apart from the communication of 3-index intermediates, which is suppressed.

1	1	10	13	1	13	1	8	8	10	10	8	13
	2	2	11	1	2	1	2	9	9	11	11	9
		3	3	12	2	3	2	3	10	10	12	12
			4	4	13	3	4	3	4	11	11	13
				5	5	1	4	5	4	5	12	12
					6	6	2	5	6	5	6	13
						7	7	3	6	7	6	7
							7	8	4	7	8	7
								9	9	5	8	9
									10	10	6	9
										11	11	7
											12	12
												13

Fig. 2 Distribution of the batch pairs on the participating computer nodes for the calculation with 13 CPUs. The numbers in the scheme denote the indices of the CPUs to which a batch pair (*i.e.* a block of two-electron MO integrals or double excitation amplitudes) is assigned. Highlighted are the blocks assigned to the three computer nodes.

Besides efficiency, another important point is the simplicity of the program. This is an important aspect for the maintainability of the existing source code as well as for future extensions. We note that the parallel RI-MP2 implementation described here is part of the ricc2 module of the Turbomole package which also includes, *e.g.*, the functionality to calculate electronic excitation energies and excited state gradients with the approximate coupled-cluster singles-and-doubles model CC2 and the related CIS(D_∞)and ADC(2) methods. The parallel implementation of the latter will be subject of a forthcoming publication. To keep the parallel implementation as simple as possible we used the following concept:

• The same source code for all computer nodes, with no specialized server process (single program multiple data (SPMD) model).

• The same source code for sequential and parallel implementation; the parallel case should differ from the sequential one only in extensions necessary for data communication and in the range of indices.

• ''Static'' distribution of computational tasks to the participating nodes. Since in RI-MP2 calculations the distribution of the computational tasks is strongly coupled with the distribution of large amounts of data (the different 3-index intermediates), a task distribution dynamically adapted during program execution would result in rather complex algorithms. Therefore, the work is distributed to the nodes according to static predefined schemes (*vide infra*).

• Whenever possible only one-sided communication (MPI calls) is used to keep the implementation of the communication steps as transparent as possible.

To implement the blocked distribution of occupied orbital indices and index pairs sketched above we define at the beginning of the calculation the following index sets:

$\mathcal{I}_m$: a block of occupied orbitals i assigned to node m

$\mathcal{J}_m$: merged set of the n_{blk} blocks $\mathcal{I}_{nj}$ (with $j = 1,\ldots,n_{blk}$) for which node m needs the 3-index ERIs B^i_{Qa} or calculates a contribution to Y^i_{Qa}

$\mathcal{S}_m$: the set of all columns in the blocked distribution to which node m calculates contributions (*cf.* Fig. 2)

$\mathcal{R}_m(n)$: the indices of the rows in column n assigned in this distribution to node m

Furthermore we define for the fast communication mode blocks $\mathcal{M}_m$ with $\sim N/n_{CPU}$ atomic orbitals assigned to node m. With this concept one obtains an efficient parallelization of most program parts concerned with the calculation of the RI-MP2 energy and the unrelaxed one- and two-particle density intermediates. These parts use only 3- and 2-index AO integrals and all steps that scale with $\mathcal{O}(\mathcal{N}^4)$ or $\mathcal{O}(\mathcal{N}^5)$ involve at least one occupied index.

A. Calculation of the three-index integral intermediates

As pointed out above, each node needs access to $n_{blk} \approx \sqrt{n_{CPU}}$ blocks of the 3-index integral intermediates B^i_{Qp} in the $\mathcal{N}^5$ scaling steps. Each node evaluates either all integrals needed there later on or, if the interconnection is fast enough, evaluates

 This journal is © the Owner Societies 2006

only one block of B^i_{Qp} and receives the remaining ones from other nodes through MPI calls or a global file system.

In the first case (slow communication, left column in Table 1) the B^i_{Qa} intermediates can be calculated in the same way as in the sequential case with the only change being that the occupied orbitals are restricted to the set $\mathscr{I}_m$. We use here the multipass algorithm from Weigend and Häser,[31] but with improved memory management, screening and symmetry treatment to reduce the number of required passes over the AO integrals and the operation count for the transformations. The calculation of B^i_{Qp} requires in this case a formal operation count of $\mathcal{O}(N^3)$ for the AO integrals plus $\mathcal{O}(nN^3)/\sqrt{n_{\mathrm{CPU}}}$ for the transformation steps, but no communication.

In the second case (fast communication, right column in Table 1) we use instead a batched I/O algorithm consisting of two parts: First, the partially transformed integrals $(P|\mu i)$ are computed and stored on disk in a parallelized loop over shells of atomic orbitals μ. These are then redistributed among the computer nodes such that the remaining partial transformations can be carried out in a parallelized loop over occupied orbitals i. Finally, each block of B^i_{Qp} is sent to all nodes on which it is needed. With this scheme one achieves, for all steps of the integral evaluation, an operation count that scales with $1/n_{\mathrm{CPU}}$ at the expense of a communication demand per node of $nNN_x/\sqrt{n_{\mathrm{CPU}}}$ for the distribution of B^i_{Qp}. (For the redistribution of the partially transformed integrals $(P|\mu i)$ the communication scales with nNN_x/n_{CPU}.)

In the I/O algorithm the symmetry of the AO integrals $(P|\mu\nu)$ with respect to the orbital indices μ and ν is not exploited. Nevertheless, it outperforms the multipass algorithm if the number of passes becomes large. Therefore the program also switches in the sequential case and slow communication mode to the I/O algorithm if the number passes exceeds a certain limit (5 by default). We further note that by the communication of $(P|\mu i)$ in the fast communication mode only the costs for the AO integral evaluation are reduced. For large molecules and small basis sets without diffuse functions, where integral screening greatly reduces the number of AO integrals that are actually calculated, it can become more efficient to evaluate the complete set of 3-index AO integrals on each node instead of communicating the $(P|\mu i)$ intermediate, which does not become sparse.

B. Parallel implementation of the $\mathcal{N}^5$-scaling steps

With the index sets defined at the beginning of section III all $\mathcal{N}^5$-scaling steps in the calculation of RI-MP2 energies and gradients can be carried out in essentially the same way as in the sequential case. Some complications occur due to the non-canonical calculation of the one-electron density matrix elements D^{MP2}_{ij} for nearly degenerate occupied orbitals, which involve the re-evaluation of some amplitudes and thereby some integrals, which in the parallel case might otherwise not be needed on the respective node. The related indices have to be identified in advance and received through MPI calls from the nodes on which they have been calculated. For this we define

$K(i)$: the set of occupied orbitals k with $|\varepsilon_k - \varepsilon_i| < t_{\mathrm{can}}$

K_m: the set of all $k \in K(i)$ with $i \in \mathscr{J}_m$ but $k \notin \mathscr{J}_m$

With this, all complications due to the parallelization are moved into the preparation and post-processing steps such that the efficiency of the time-determining steps is not diminished by delays in the communication. The $\mathcal{N}^5$-scaling steps for the calculation of the Y^i_{Qa} intermediate and the density matrix elements evaluated according to the non-canonical formula can be sketched as:

loop $n \in \mathscr{S}_m$, loop I (where $I \subseteq \mathscr{I}_n$)
read B^i_{Qa} for all $i \in I$
loop $n' \in \mathscr{R}_m(n)$, loop $j \in \mathscr{I}_{n'}$ with $j \leq i$
  * read B^j_{Qb}
  * $t^{ij}_{ab} \leftarrow B^i_{Qa}\, B^j_{Qb}/\{\varepsilon_i - \varepsilon_a + \varepsilon_j - \varepsilon_b\}$
  * $Y^i_{Pa} \leftarrow (2t^{ij}_{ab} - t^{ij}_{ab})\, B^j_{Pb}$ and for $j \neq i$ also $Y^j_{Pb} \leftarrow (2t^{ij}_{ba} - t^{ij}_{ba})\, B^i_{Pa}$
  * $D^{\mathrm{MP2}}_{ab} \leftarrow (1 + (1 - \delta_{ij})\hat{P}_{ij})\,(2t^{ij}_{ac} - t^{ij}_{ca})t^{ij}_{bc}$ for $a \geq b$
  * $D^{\mathrm{MP2}}_{ii} \leftarrow -(2t^{ij}_{ab} - t^{ij}_{ba})\, t^{ij}_{ab}$ and for $j \neq i$ also $D^{\mathrm{MP2}}_{jj} \leftarrow -(2t^{ij}_{ab} - t^{ij}_{ba})\, t^{ij}_{ab}$
  * loop $k \in K(i)$
  • read B^k_{Qb}
  • $t^{kj}_{ab} \leftarrow B^k_{Qa}\, B^j_{Qb}/\{\varepsilon_k - \varepsilon_a + \varepsilon_j - \varepsilon_b\}$
  • $D^{\mathrm{MP2}}_{ki} \leftarrow -(2t^{ij}_{ab} - t^{ij}_{ba})\, t^{ki}_{ab}$
  * end loop k
  * for $j \neq i$ loop $k \in K(j)$
  • read B^k_{Qb}
  • $t^{ki}_{ab} \leftarrow B^k_{Qa}\, B^i_{Qb}/\{\varepsilon_k - \varepsilon_a + \varepsilon_i - \varepsilon_b\}$
  • $D^{\mathrm{MP2}}_{kj} \leftarrow -(2t^{ii}_{ab} - t^{ji}_{ba})\, t^{ki}_{ab}$
  * end loop k
end loop n', loop j
store Y^i_{Pa} and Y^j_{Pb} on disk (distributed)
end loop n, loop I

If only the RI-MP2 energy is needed, it can be evaluated directly after the calculation of the integrals $(ia|jb)$ according to eqn (1) and the calculation of Y^i_{Pa} and the density matrix elements D^{MP2}_{pq} can be skipped. If the latter intermediates are needed, the contributions to the Y^i_{Pa} intermediate are added and redistributed such that each node has the complete results for Y^i_{Pa} for all $i \in \mathscr{J}_m$ (requiring the communication of $\sim 2nN\,N_x/\sqrt{n_{\mathrm{CPU}}}$ floating point numbers per node).

C. Evaluation of the relaxed densities and the gradient contributions

In the next step the intermediates $\Gamma_{Pi\nu}$ (eqn (11)), γ_{PQ} (eqn (9)), and L''_{aq} (eqn (20)) are computed from the Y intermediates. These operations are (as in the sequential case[31,35]) organized in a loop over occupied orbitals and easily parallelized by restricting the set of occupied orbitals on each node to the index sets $\mathscr{I}_m$ defined at the beginning of section III.

Somewhat more involved is the parallel calculation of the matrix L_{iq} (eqn (19)) and the 3-index two-electron density in the AO basis, $\Delta^{\mathrm{ao}}_{P,\mu\nu}$ (eqn (8)). Similar to the evaluation of the 3-index integral intermediate B^i_{Qa}, these steps require either the communication of a 3-index array (here $\Gamma_{Pi\mu}$) between the nodes or that the complete set of AO (derivative) 3-index ERIs is calculated on each node. As described in Table 2 we use the latter strategy only if the interconnection between the computer nodes is too slow for an efficient communication of 3-index intermediates. If the interconnection is fast enough the intermediate $\Gamma_{Pi\mu}$ is redistributed among the nodes such that

Table 1 Algorithm for the evaluation of the 3-index integral intermediates B^i_{Qp}. For simplicity the screening and symmetry treatment are not described

Scheme for slow communication	Scheme for fast communication
Loop I (where $I \subseteq \mathscr{I}_m$) Evaluate $(P\|\mu\nu)$ for all $\mu \geq \nu$, P $B^i_{Qp} \leftarrow C_{\mu p} C_{\nu i} V^{-1/2}_{PQ}\ (P\|\mu\nu)$ for $i \in I$ Store B^i_{Qp} on disk (distributed) End loop I	Loop M (where $M \subseteq \mathscr{M}_m$) Evaluate $(P\|\mu\nu)$for $\mu \in M$ $(P\|\mu i) \leftarrow C_{\nu i}(P\|\mu\nu)$ for $\mu \in M$, all i Store $(P\|\mu i)$ on disk (distributed) End loop M Receive $(P\|\mu i)$ for all $i \in \mathscr{I}_m$, $\mu \notin \mathscr{M}_m$ $B^i_{Qp} \leftarrow V^{-1/2}_{PQ}\ C_{\mu p}(P\|\mu i)$ for $i \in \mathscr{I}_m$ store B^i_{Qp} on disk (distributed) Receive B^i_{Qa} for all $i \in \mathscr{J}_m/\mathscr{I}_m$
Operation count per node: $\propto \mathscr{O}(N^2 N_x) + \mathscr{O}(nN^2 N_x)/\sqrt{n_{CPU}}$ No communication	Operation count per node: $\mathscr{O}(N^2 N_x)/n_{CPU} + \mathscr{O}(nN^2 N_x)/n_{CPU}$ Communication count per node: $\propto nNN_x \times \left(\frac{1}{n_{CPU}} + \frac{1}{\sqrt{n_{CPU}}}\right)$

calculation of L_{iq} and the gradient contribution of the 3-index integrals can be parallelized over AO shells.

Once the density matrix D^{MP2}_{pq} and the intermediates L_{iq} and L''_{aq} are available, the right-hand side vector η^κ is set up, the Z-vector equations are solved for the Lagrange multipliers $\bar{\kappa}$ and the matrices D^{eff} and F^{eff} are computed. The computationally demanding steps of this part and the evaluation of the remaining gradient contributions are the contraction of density matrices or, in the solution of the Z-vector equations, trial vectors with the CPHF matrix A_{pqrs} and the calculation of the 4-index derivative integrals $(\mu\nu|\kappa\lambda)^{[x]}$. The contractions of the CPHF matrix with densities or trial vectors are formulated as Fock matrix builds and the $(\mu\nu|\kappa\lambda)^{[x]}$ are contracted immediately with the separable part of the two-electron density $d^{sep,ao}$ which is constructed on-the-fly from the one-electron densities D^{SCF} and D^{eff}. Both steps can be parallelized straightforwardly over pairs of AO shells in a similar way to parallel implementations of conventional MP2 or SCF gradients. We make use of the routines from the dscf and grad programs of the Turbomole package[5,16,27,36] for this part, but use, in contrast to the parallel SCF (gradient) implementation of Turbomole, a static task distribution, *i.e.* the assignment of AO shell pairs to nodes is not changed *e.g.* between the iterations for the Z-vector equations. The latter can be carried out either fully integral-direct or semi-direct similar to ground state SCF calculations with the dscf program. These steps involve only the communication of one-electron densities and a few other 2-index intermediates and thus we use the same algorithms irrespective of the speed of the network interconnection.

IV. Performance of parallel RI-MP2 energy and gradient calculations

To evaluate the efficiency of the parallel implementation of the analytic RI-MP2 gradient we choose four typical test systems with the structures shown in Fig. 3:

- 1-*tert*-butyl-6-cyano-1,2,3,4-tetrahydroquinoline (NTC6), a planar alkylaminobenzonitrile,[37,38] in the TZVPP basis set[6] (748 orbital and 1756 auxiliary basis functions, 84 correlated electrons, C_1 symmetry).
- A calicheamicin model taken from ref. 29, which has also no point group symmetry. These calculations have been done in the cc-pVTZ basis sets[39–41] with 934 orbital and 2429 auxiliary functions and 124 electrons have been correlated.
- The fullerene C_{60}, which has I_h symmetry, but since the present version of the parallel implementation can handle only

Table 2 Algorithm for the parallel calculation of the matrix L_{iq} defined in eqn (19) and the contribution of the 3-index derivative ERIs $(P|\mu\nu)^{[x]}$ to the gradient

Scheme for slow communication	Scheme for fast communication
Loop M (where M is a subset of AO shells) Read $\Gamma_{Pi\mu}$ for $\mu \in M$ and $i \in \mathscr{I}_m$ For calculation of L_{iq}: — Evaluate $(P\|\mu\nu)$ for $\mu \in M$ — $L_{i\nu} \leftarrow L_{i\nu} + (P\|\mu\nu)\ \Gamma_{Pi\mu}$ for $i \in \mathscr{I}_m$ For gradient contribution: — $\Delta^{ao}_{P,\mu\nu} \leftarrow \Sigma_{i\in\mathscr{I}_m} \Gamma_{Pi\mu}\ C_{i\nu}$ for $\mu \in M$ — Evaluate $(P\|\mu\nu)^{[x]}$ for $\mu \in M$ — $g_x \leftarrow g_x + \Delta^{ao}_{P,\mu\nu}\ (P\|\mu\nu)^{[x]}$ End loop M For L_{iq}: $L_{iq} = L_{i\nu} C_{\nu q}$ Operation count per node: $\propto \mathscr{O}(N^2 N_x) + \mathscr{O}(nN^2 N_x)/n_{CPU}$ No communication	Receive $\Gamma_{Pi\mu}$ for all $i \notin \mathscr{I}_m$, $\mu \in \mathscr{M}_m$ Loop M (where $M \subseteq \mathscr{M}_m$) Read $\Gamma_{Pi\mu}$ for $\mu \in M$, all i For calculation of L_{iq}: — Evaluate $(P\|\mu\nu)$ for $\mu \in M$ — $L_{i\nu} \leftarrow L_{i\nu} + (P\|\mu\nu)\ \Gamma_{Pi\mu}$ for all i For gradient contribution: $\Delta^{ao}_{P,\mu\nu} \leftarrow \Gamma_{Pi\mu} C_{i\nu}$ for $\mu \in M$, all i — Evaluate $(P\|\mu\nu)^{[x]}$ for $\mu \in M$ — $g_x \leftarrow g_x + \Delta^{ao}_{P,\mu\nu}\ (P\|\mu\nu)^{[x]}$ End loop M For L_{iq}: $L_{iq} = L_{i\nu} C_{\nu q}$ Operation count per node: $\propto \mathscr{O}(N^2 N_x)/n_{CPU} + \mathscr{O}(nN^2 N_x)/n_{CPU}$ Communication count per node: $\propto nNN_x/n_{CPU}$

This journal is © the Owner Societies 2006

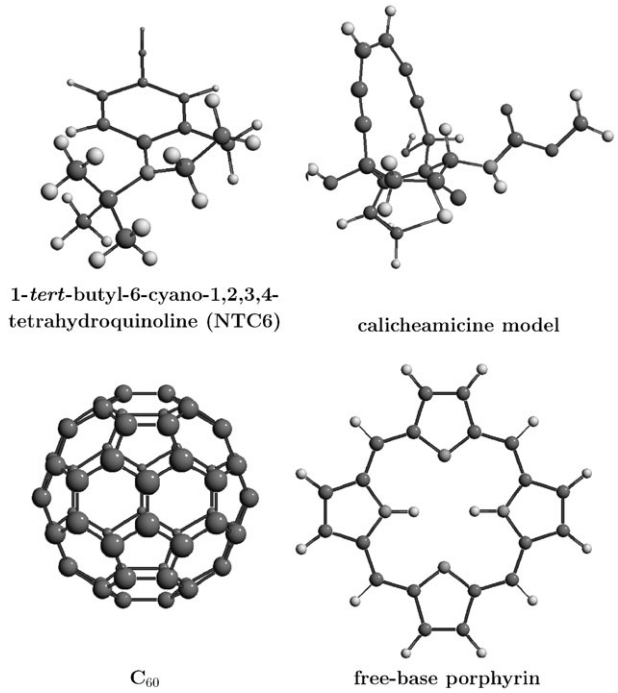


Fig. 3 Structures of the four test examples used to benchmark the performance of parallel RI-MP2 gradient calculations. For the details of the basis sets and the number of correlated electrons see text.

Abelian point groups in the evaluation of the MP2 density the calculations have been carried out in the subgroup D_{2h}. The cc-pVTZ basis set has been used, which in this case comprises 1800 orbital and 4860 auxiliary basis functions and the 240 valence electrons were correlated.

• Free-base porphyrin (D_{2h} symmetry) in the cc-pVTZ basis (916 orbital and 2364 auxiliary basis functions, 114 correlated electrons).

To benchmark the calculation of MP2 energies we used the largest of the above examples—the calicheamicin model and the C_{60} fullerene—and in addition:

• A cluster of 40 water molecules as an example of a system where integral prescreening leads to large reductions of the costs in conventional MP2 calculations. The basis sets are 6-31G* for the orbital[42] and cc-pVDZ for the auxiliary[7] basis with 760 and 3840 functions, respectively; the point group is C_1 and the 320 valence electrons have been correlated.

• A chlorophyll derivative shown in Fig. 4 which has also no point group symmetry. The cc-pVDZ basis with 918 orbital and 3436 auxiliary functions have been used and 264 electrons have been correlated.

The maximum amount of core memory used by the program was, in all of the calculations, limited to 750 MB. The calculations were run on two different Linux clusters: one cluster with *ca.* 100 Xeon Dual 2.8 GHz nodes connected through a cascaded Gigabit network and a second cluster with *ca.* 64 Athlon 1800MP MHz nodes connected through a 100 MBit fast Ethernet network. Due to a much larger load on the first cluster and its network the transfer rates reached in the benchmark calculations varied between *ca.* 80–200 Mbit s^{-1} per node. On the Athlon Cluster with the 100 Mbit

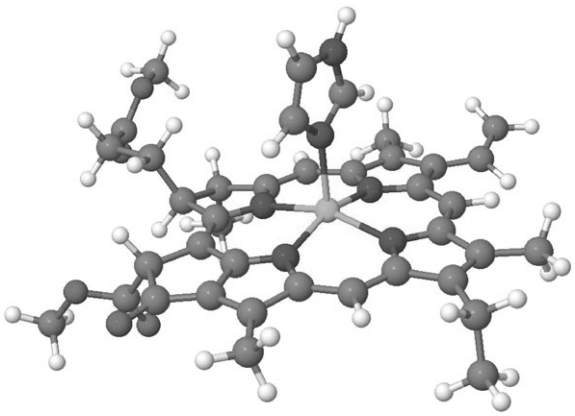


Fig. 4 Structure of the chlorophyll derivative used to benchmark the performance of parallel RI-MP2 energy calculations. For the basis set used and the number of active electrons see text.

network we reached transfer rates of *ca.* 20–50 Mbit s^{-1} per node.

Although geometry optimizations with the sequential implementation would for all four test examples be tedious (but still feasible) calculations, the time-determining step in these examples is still the solution of the Z-vector or CPHF equations, followed by the $\mathcal{O}(\mathcal{N}^5)$-scaling steps for the unrelaxed MP2 density and the Y intermediate and the calculation of the two-electron derivative integrals. This is related to the fact that for RI-MP2 based on conventional (non-RI) Hartree–Fock the computational costs for the direct solution of the CPHF equations are similar to those of a direct SCF calculation. We note at this point that the present version of the program includes an option to compute RI-SCF based gradients, but for the present examples (*i.e.* size of the molecules and chosen basis sets) this leads to only moderate savings. Therefore, we have used the RI-CPHF option for these calculations only for the preoptimization of the Z-vector. The (4-index ERI based) CPHF equations were solved in a fully integral-direct way for free-base porphyrin and NTC6 and a semi-direct scheme was used for C_{60} and calicheamicin (storing at most 10 and 12 GB of integrals per node, respectively).

Fig. 5 shows timings for the calculation of MP2 energies for the C_{60} example. On both architectures in sequential runs about 55% of the time is spent in the matrix multiplication for the $\mathcal{N}^5$ step. With increasing numbers of nodes this contribution slowly decreases. In case of the "slow communication" mode this is due to the fact that the integral evaluation takes an increasing fraction of the total wall time, while in the "fast communication" mode (and here in particular on the cluster with the slower network) it is due to the increasing fraction of time spent in the communication of the 3-index MO integral intermediate B^i_{Qa}. Steps that are not parallelized—such as *e.g.* the evaluation of the matrix V_{PQ} of 2-index ERIs, its Cholesky decomposition and formation of the inverse—take only a marginal fraction of the total wall time and the fraction of the time spent in the I/O stays approximately constant with the number of nodes used for the calculation. Another important message from Fig. 5 is that even with a relatively slow network it is advantageous to communicate the 3-index intermediates,

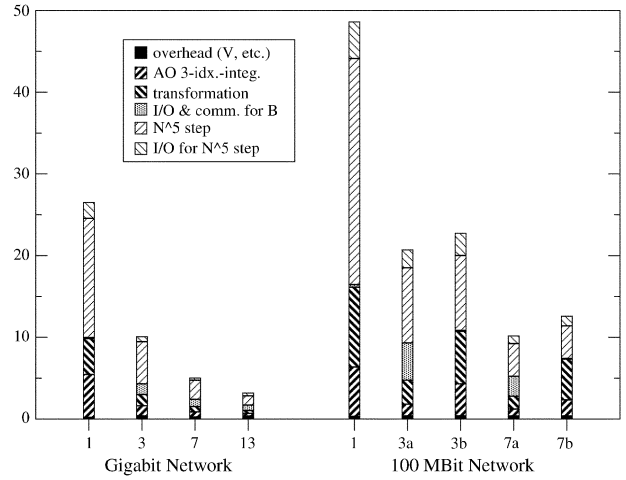


Fig. 5 Timings for the most important steps in parallel RI-MP2 energy calculations for C_{60} in the cc-pVTZ basis (240 electrons correlated). See text for technical details of the machines used. At the abscissa we indicate the number of CPUs used for the calculations. For the cluster with a 100 Mbit network letters "a" and "b" are added for calculations in the "fast" and "slow" communication modes, respectively. On the other cluster only the former program mode has been tested. The fraction denoted "overhead" includes most non-parallel steps, as the calculation of the Coulomb metric V and the inverse of its Cholesky decomposition, I/O and communication of MO coefficients, *etc*. With "AO 3-idx.-integ" we denote the time spent for the calculation of the AO 3-index integrals $(P|\mu\nu)$ and with "transformation" and "I/O & comm. for B" the fractions spent in the 3-index transformations for the intermediates B^i_{Qa} and for saving these intermediates on disk and/or distributing them to other computer nodes. "N^5 step" and "I/O for N^5 step" are the fractions spent in the $\mathcal{N}^5$-scaling matrix multiplication and the I/O of B intermediates, respectively, during the calculation of two-electron MO integrals. For parallel calculations idle times caused by non-perfect load balance are included under the point "I/O for N^5 step".

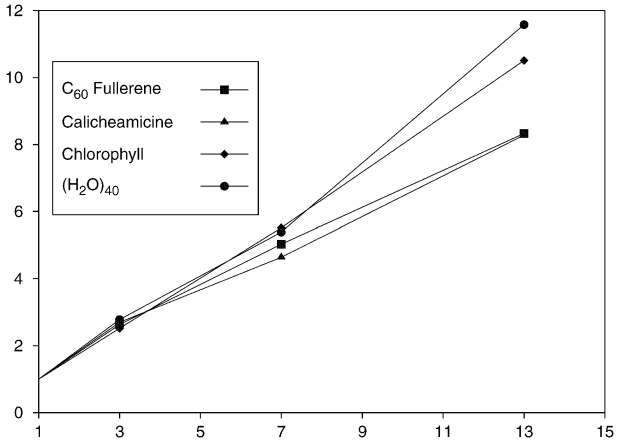


Fig. 6 Speedup obtained for parallel RI-MP2 energy calculations on the Linux cluster with the Gigabit network for the four test examples. The number of nodes is given on the abscissa and the speedup (defined as the wall time of the parallel calculation divided by the wall time of the sequential run) is indicated on the ordinate.

although on the cluster with the slower network the difference in performance between the two modes is not large. We note, however, that this also depends on the size of the system and the basis sets.

Because of the symmetry of the molecule, an RI-MP2 energy calculation for C_{60} is not really a large scale application today. The same holds for the other three test examples. Nevertheless, for these (for parallel calculations) small examples the speedup obtained with the present implementation are reasonable as Fig. 6 shows. The speedup obtained increases with the system size as the computational costs become dominated by the $\mathcal{N}^5$-scaling matrix multiplication in the chosen construction of the MO 4-index ERIs and the less efficient parallelizing calculation and/or communication of the 3-index MO integrals becomes unimportant for the total wall time. Therefore, the largest speedups are obtained for the chlorophyll derivative and the $(H_2O)_{40}$ cluster.

Of the four examples used to benchmark the parallel evaluation of MP2 gradients, only the C_{60} fullerene required the use of a non-canonical calculation of density matrix elements for closely degenerate orbitals. This calculation was run in the D_{2h} subgroup of the molecular point group I_h. In this subgroup, 16 of the total 990 symmetry-allowed elements of D_{ij} had an energy separation between the occupied orbitals below the threshold value (chosen as 10^{-3} au). The calculation of these matrix elements took between 5% (sequential) and 12% (13 nodes) of the time spent in the $\mathcal{N}^5$-scaling part and thus played an unimportant role in the total wall time of the calculations, which was dominated by the time needed to solve the CPHF equations. Fig. 7 shows an overview of the timings for the most important steps in the RI-MP2 gradient

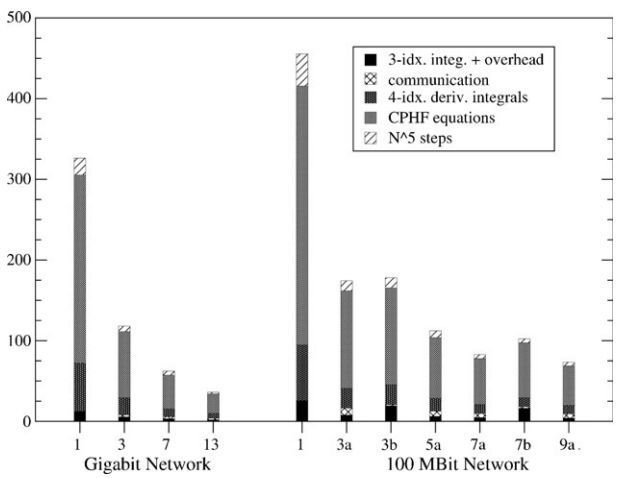


Fig. 7 Timings for the most important steps in parallel RI-MP2 gradient calculations for NTC6 in the TZVPP basis (84 electrons correlated). See text for technical details of the machines used and for the explanation of the abscissa (caption of Fig. 5). The fraction "3-idx. integr + overhead" comprises the non-parallel steps and the calculation of the B^i_{Qp} intermediates, "communication" denotes the time spent in the communication of integrals and intermediates. The labels "4-idx. deriv. integrals" and "CPHF equations" denote the time spent in the evaluation of the gradient contribution from 4-index AO derivative integrals and the solution of the Z vector equations (including the construction of the right hand side), respectively. The point "N^5 steps" includes the $\mathcal{N}^5$-scaling matrix multiplications for the calculation of 4-index MO integrals, the Y intermediate, and the unrelaxed MP2 density matrix, as well as the I/O of B and Y intermediates required for these steps.

 This journal is © the Owner Societies 2006

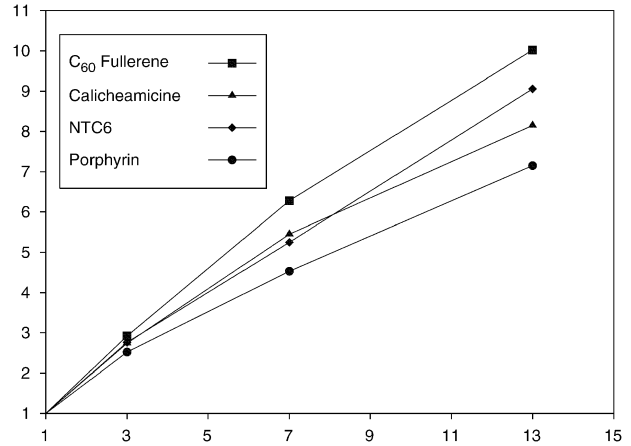


Fig. 8 Speedup obtained for parallel RI-MP2 gradient calculations on the Linux cluster with a Gigabit network for the four test examples. The number of nodes is given on the abscissa and the speedup (defined as the wall time of parallel calculation divided by the wall time of the sequential run) is indicated on the ordinate.

calculation and their performance on the two Linux clusters used for the test calculations with an increasing number of nodes. As observed for RI-MP2 energy calculations, there is little difference in performance between the modes for fast and slow communication when using the slower network. However, as noted above, the relative speeds of the two program modes also depend on the structure of the molecule and the basis sets used, but as the total time required for RI-MP2 gradient calculations is mostly dominated by the solution of the CPHF equations, which do not involve the communication of integral intermediates, the communication becomes less important than for applications which focus on the RI-MP2 energy only. The performance obtained for the four test examples is shown in Fig. 8.

V. First application: the fullerenes C_{60} and C_{240}

An important aspect of the parallel implementation of RI-MP2 gradients is that it allows the combination of the fast RI-MP2 approach with *parallel* Hartree–Fock self-consistent field (HF-SCF) calculations, available today in many program packages for electronic structure calculations, to optimize geometries for relatively large molecules at the MP2 level. Since this paper focuses on the description of the implementation we give only one example for such an application: the C_{60} fullerene which also served as a prototype example above. The ground state equilibrium structure of C_{60} has been studied at the MP2 level some years ago by Häser and Almlöf,[43] but due to the large computational costs of MP2 the calculations had to be limited to a singly polarized TZP basis set ([5s3p1d], 1140 basis functions), which is known to cover only about 75% of the correlation energy. We could now repeat this calculation using the (doubly polarized) correlation consistent triple-ζ basis cc-pVTZ ([4s3p2d1f], 1800 basis functions), which typically gives correlation energies almost within 90% of the basis set limit, and cc-pVQZ ([5s4p3d2f1g], 3300 basis functions), which usually cuts the remaining basis set errors into half again. The structure optimization has been run on 7 nodes. On average, about 20–30% of the (wall) time was spent in the HF-SCF and 70–80% in the RI-MP2 gradient calculation. The results for the bond lengths and the total energies are summarized in Table 3 together with the results from ref. 43 and the available experimental data. As anticipated from the quality of the basis sets, the result for the correlation energy increases by about 15% from the MP2/TZP to the MP2/cc-pVTZ calculation and again by about 6% from the cc-pVTZ to the cc-pVQZ basis. The changes in the bond lengths from the MP2/TZP to the MP2/cc-pVQZ level are within 0.004–0.005 Å of the same magnitude as those between the MP2/DZP and MP2/TZP calculations. However, the difference between the two C–C distances remains almost unchanged, and the comparison with the experimental data is also unaffected, since the error bars of the latter are within about ± 1 pm of the same order of magnitude as the basis effects. We can conclude that the present results are very close to the MP2 (valence) basis set limit. Core correlation effects will lead to a further slight contraction of the bond lengths, but the largest uncertainty comes from higher-order correlation effects which

Table 3 Equilibrium bond distances of C_{60}; d_{C-C} denotes the distance between adjacent C atoms in a five-ring and $d_{C=C}$ denotes the distance between the C–C bond shared between two six-rings. The bond distances are given in Å and the total energies in E_h

Method	d_{C-C}	$d_{C=C}$	Energy
SCF/DZP[a]	1.450	1.375	−2272.10290
SCF/TZP[a]	1.448	1.370	−2272.33262
MP2/DZP[b]	1.451	1.412	−2279.73496
MP2/TZP[b]	1.446	1.406	−2280.41073
MP2/cc-pVTZ[c]	1.443	1.404	−2281.65632
MP2/cc-pVQZ[c]	1.441	1.402	−2282.34442
Exp.[d]	1.458(6)	1.401(10)	
Exp.[e]	1.45	1.40	
Exp.[f]	1.432(9)	1.388(5)	

[a] From ref. 44. [b] From ref. 43. [c] This work. The SCF energies for the MP2 optimized structures are −2272.404 H (cc-pVTZ) and −2272.536 H (cc-pVQZ). [d] Gas phase electron diffraction, ref. 45. [e] Solid state NMR, ref. 46. [f] X-ray structure of $C_{60}(OsO_4)(4\text{-}tert\text{-butylpyridine})_2$, ref. 47.

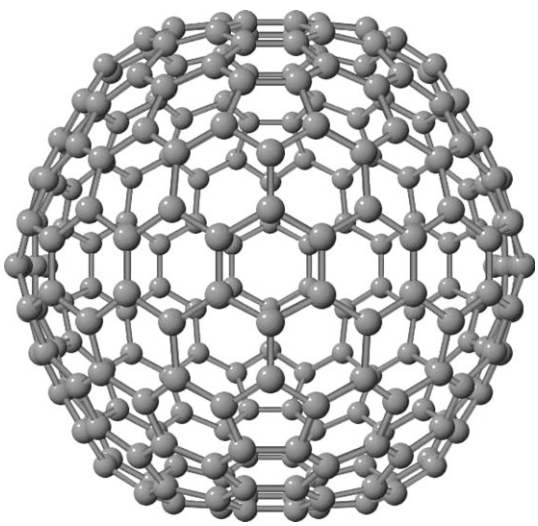


Fig. 9 Structure of the icosahedral C_{240} fullerene.

will probably increase the bond lengths in this system, however probably no more than 0.005 Å. Therefore we estimate that the present results for the equilibrium bond distances (r_e) of the Buckminster fullerene C_{60} are accurate within ±0.005 Å. This is slightly less than the uncertainty of the presently available experimental data. Within their uncertainties the *ab initio* calculations and the experiments are in good agreement.

An example demonstrating how the parallel implementation extends the applicability of canonical MP2 calculations to larger molecules is the next larger icosahedral homologue of the Buckminster fullerene C_{60}, the C_{240} molecule (Fig. 9). The correlation consistent triple-ζ basis cc-pVTZ for this molecule comprises 7200 basis functions and, if the 1s core orbitals are kept frozen, 960 electrons have to be correlated. We ran this calculation on the above mentioned Linux cluster with Dual Xeon 2.8 GHz nodes connected by a Gigabit network. Because in our implementation the memory demands increase for non-Abelian point groups with the square of the dimension of the irreducible representations, the calculation was carried out in the D_{2h} subgroup of the molecular point group I_h. On 19 CPUs the RI-MP2 calculation was completed after 16 h 6 min. About 12.5% of the time was spent in the evaluation and distribution of the 2- and 3-index integrals and 85% on the $\mathcal{O}(n^2N^3)$ scaling construction of the 4-index integrals in the MO basis $(ia|jb)$ in eqn (5). In D_{2h} symmetry about 6×10^{11} 4-index MO integrals (≈ 4.8 TByte) had to be evaluated to calculate the MP2 energy. This shows that such a calculation would, with a conventional (non-RI) MP2, require either an enormous amount of disk space or many costly re-evaluations of the 4-index AO two-electron integrals and would be difficult to carry out even on a massively parallel architecture. To the best of our knowledge this is the largest canonical MP2 calculation done to date. With the parallel implementation of the RI-MP2 approach calculations of this size can now be carried out on PC clusters built with standard (and thus low cost) hardware and are expected to become routine applications soon.

The total energy of C_{240} obtained with MP2/cc-pVTZ at the BP86[48–50] /SVP[51,52] optimized structure[53] is −9128.832 558 H. For the Buckminster fullerene C_{60} a single point MP2/cc-pVTZ calculation at the BP86/SVP optimized geometry gives a total energy of −2281.645 107 H. Neglecting differential zero-point energy effects, which in this case are expected to be small, we obtain from our calculations an estimate for the reaction enthalpy of $4 \times C_{60} \rightarrow C_{240}$ of −2.25 H, *i.e.* a change in the enthalpy of formation per carbon atom of −9.4 mH or −25 kJ mol^{-1}. This can be compared with the experimental result[54] for $\Delta_f H^0$ of C_{60} relative to graphite of 39.25 ± 0.25 kJ mol^{-1}. Thus, the present calculations predict that the strain energy per carbon atom in C_{240} is ≈ 15 kJ mol^{-1}, which is only about 35% of the respective value in C_{60}.

VI. Summary and conclusions

We presented a parallel implementation of RI-MP2 energies and gradients for distributed memory architectures. The implementation is based on canonical MP2 equations and the aim was for it to be applied to large molecules with medium sized basis sets, and in particular such systems where local correlation methods can not efficiently be applied, such as molecules with large conjugated subgroups and long range correlation effects. An important motivation for this is that the present parallel MP2 implementation is intended as a starting point for a future extension to second-order response methods for excited states.

The chosen parallelization strategy is designed to obtain high efficiency in the time-determining steps, while conserving the comprehensible structure of the underlying sequential code to allow for future generalizations to other methods. For large systems the computational costs in RI-MP2 energy calculations are dominated by the $\mathcal{O}(n^2N^3)$-scaling steps, which have been parallelized over pairs of occupied orbitals. In gradient calculations for medium sized systems or with basis sets containing diffuse functions the (formally) $\mathcal{O}(N^4)$-scaling steps (Z-vector equations and the contribution of derivatives of 4-index ERIs) dominate the computational costs. These are parallelized over pairs of atomic orbital shells in a similar way to that described for parallel SCF codes.[5,36] For both parts the efficiency of the parallelization is limited by the load balancing, which for the $\mathcal{O}(n^2N^3)$-scaling part is determined by the distribution of the occupied orbitals in n_{CPU} equally sized subsets and for the $\mathcal{O}(N^4)$-scaling part by differences in the integral costs and effects of integral prescreening which are only roughly accounted for in the task distribution.

A conflict appears in the next most expensive steps which scale with $\mathcal{O}(nN^3)$ and $\mathcal{O}(N^3)$ where a (theoretically) linear speed increase with n_{CPU} can only be achieved if communications demands of $\mathcal{O}(nN^2)$ are accepted. Otherwise the speedup that can be reached for the steps scaling with $\mathcal{O}(nN^3)$ is limited by $\approx \sqrt{n_{CPU}}$ and the AO 3-index integral calculation scaling with $\mathcal{O}(N^3)$ can not be parallelized. Since the optimal choice depends on the hardware—in particular the bandwidth for the network interconnection between the nodes and the availability of a fast global file system—the investigated system, and the basis sets used, we accounted for this conflict by devising (and implementing) two alternative schemes for slow and fast communication. In the former the communication of 3-index intermediates is mostly avoided at the expense of a reduced speedup for the calculation of 3-index two-electron integrals. In the latter case all parts scaling with $\mathcal{O}(\mathcal{N}^4)$ and most $\mathcal{O}(\mathcal{N}^3)$ steps are parallelized such that (in principle) a linear speedup with n_{CPU} could be obtained.

We demonstrated for a number of test examples that with the implemented parallelization strategy good speedups are obtained on Linux PC clusters with standard hardware and Fast Ethernet or Gigabit networks, provided that the number of correlated occupied orbitals, n, is large compared to the number of computer nodes, n_{CPU}, used for the calculation. (We recommend $n/n_{CPU} \geq 10$.) The efficiency of the parallel implementation increases with the system size and the program is thus well suited for large scale applications. With two applications—a geometry optimization for C_{60} at the MP2/cc-pVQZ level (3300 basis functions) and a MP2/cc-pVTZ energy calculation for C_{240} (7200 basis functions)—we showed how the parallel implementation extends the applicability of canonical MP2 to larger systems. The developed parallel RI-MP2 code is part of the ricc2 module of the Turbomole package[16,17] and will be made available in future releases of this package.

This journal is © the Owner Societies 2006

Acknowledgements

The authors are indebted to R. Ahlrichs for many fruitful discussions and hints. We thank F. Furche for providing the BP86/SVP optimized geometry of C_{240}. The present work has been supported by the Deutsche Forschungsgemeinschaft (DFG) through project HA 2588/2.

References

1 J. L. Whitten, *J. Chem. Phys.*, 1973, **58**, 4496.
2 B. I. Dunlap, J. W. D. Connolly and J. R. Sabin, *J. Chem. Phys.*, 1979, **71**, 3396.
3 O. Vahtras, J. E. Almlöf and M. W. Feyereisen, *Chem. Phys. Lett.*, 1993, **213**, 514.
4 K. Eichkorn, O. Treutler, H. Öhm, M. Häser and R. Ahlrichs, *Chem. Phys. Lett.*, 1995, **242**, 652.
5 M. von Arnim and R. Ahlrichs, *J. Comput. Chem.*, 1998, **19**, 1746.
6 F. Weigend, M. Häser, H. Patzelt and R. Ahlrichs, *Chem. Phys. Lett.*, 1998, **294**, 143.
7 F. Weigend, A. Köhn and C. Hättig, *J. Chem. Phys.*, 2001, **116**, 3175.
8 C. Hättig, *Phys. Chem. Chem. Phys.*, 2005, **7**, 59.
9 H.-J. Werner, F. R. Manby and P. J. Knowles, *J. Chem. Phys.*, 2003, **118**, 8149.
10 M. Schütz, H.-J. Werner, R. Lindh and F. R. Manby, *J. Chem. Phys.*, 2004, **121**, 737.
11 F. R. Manby, *J. Chem. Phys.*, 2003, **119**, 4607.
12 S. Ten-no and F. R. Manby, *J. Chem. Phys.*, 2003, **119**, 5358.
13 D. E. Bernholdt and R. J. Harrison, *Chem. Phys. Lett.*, 1996, **250**, 477.
14 D. E. Bernholdt, *Parallel Comput.*, 2000, **26**, 945.
15 F. Weigend and M. Häser, *Theor. Chem. Acc.*, 1997, **97**, 331.
16 R. Ahlrichs, M. Bär, M. Häser, H. Horn and C. Kölmel, *Chem. Phys. Lett.*, 1989, **162**, 165.
17 TURBOMOLE, Program Package for *ab initio* Electronic Structure Calculations, http://www.turbomole.com.
18 A. Köhn and C. Hättig, *Chem. Phys. Lett.*, 2002, **358**, 350.
19 T. Korona and H. J. Werner, *J. Chem. Phys.*, 2003, **114**, 3006.
20 T. D. Crawford and R. A. King, *Chem. Phys. Lett.*, 2002, **366**, 611.
21 M. Head-Gordon, R. J. Rico, M. Oumi and T. J. Lee, *Chem. Phys. Lett.*, 1994, **219**, 21.
22 O. Christiansen, H. Koch and P. Jørgensen, *Chem. Phys. Lett.*, 1995, **243**, 409.
23 J. Schirmer, *Phys. Rev. A*, 1981, **26**, 2395.
24 A. B. Trofimov and J. Schirmer, *J. Phys. B*, 1995, **28**, 2299.
25 J. Oddershede, *Advances in Chemical Physics, Ab initio Methods in Quantum Chemistry, Part III*, ed. K. P. Lawley, John Wiley & Sons, New York, 1987, vol. 69, pp. 201–239.
26 M. J. Packer, E. K. Dalskov, T. Enevoldsen, H. J. A. Jensen and J. Oddershede, *J. Chem. Phys.*, 1980, **105**, 5886.
27 M. Häser and R. Ahlrichs, *J. Comput. Chem.*, 1989, **10**, 104.
28 F. Haase and R. Ahlrichs, *J. Comput. Chem.*, 1993, **14**, 907.
29 P. Pulay, S. Saebø and K. Wolinski, *Chem. Phys. Lett.*, 2001, **344**, 543.
30 C. Hättig, *J. Chem. Phys.*, 2003, **118**, 7751.
31 F. Weigend and M. Häser, *Theor. Chem. Acc.*, 1997, **97**, 331.
32 A. Köhn and C. Hättig, *J. Chem. Phys.*, 2003, **119**, 5021.
33 C. Hättig, *Adv. Quantum Chem.*, 2005, **50**, 37.
34 C. Hättig and F. Weigend, *J. Chem. Phys.*, 2000, **113**, 5154.
35 C. Hättig and A. Köhn, *J. Chem. Phys.*, 2002, **117**, 6939.
36 S. Brode, H. Horn, M. Ehrig, D. Moldrup, J. E. Rice and R. Ahlrichs, *J. Comput. Chem.*, 1993, **14**, 1142.
37 K. A. Zachariasse, S. I. Druzhinin, W. Bosch and R. Machinek, *J. Am. Chem. Soc.*, 2004, **126**, 1705.
38 A. Hellweg, C. Hättig and A. Köhn, *J. Am. Chem. Soc.*, 2005, submitted for publication.
39 T. H. Dunning, *J. Chem. Phys.*, 1989, **90**, 1007.
40 R. A. Kendall, T. H. Dunning and R. J. Harrison, *J. Chem. Phys.*, 1992, **96**, 6796.
41 D. E. Woon and T. H. Dunning, *J. Chem. Phys.*, 1993, **98**, 1358.
42 W. J. Hehre, R. Ditchfield and J. A. Pople, *J. Chem. Phys.*, 1972, **56**, 2257.
43 M. Häser, J. Almlöf and G. E. Scuseria, *Chem. Phys. Lett.*, 1991, **181**, 497.
44 G. E. Scuseria, *Chem. Phys. Lett.*, 1991, **176**, 423.
45 K. Hedberg, L. Hedberg, D. S. Bethune, C. A. Brown, H. C. Dorn, R. D. Johnson and M. De Vries, *Science*, 1991, **254**, 410.
46 C. S. Yannoni, P. P. Bernier, D. S. Bethune, G. Meijer and J. R. Salem, *J. Am. Chem. Soc.*, 1991, **113**, 3190.
47 J. M. Hawkins, A. Meyer, T. A. Lewis, S. Lorin and F. J. Hollander, *Science*, 1991, **252**, 312.
48 J. P. Perdew, *Phys. Rev. B: Condens. Matter*, 1986, **33**, 8822.
49 J. P. Perdew, *Phys. Rev. B: Condens. Matter*, 1986, **34**, 7046.
50 A. D. Becke, *Phys. Rev. A: At., Mol., Opt. Phys.*, 1988, **38**, 3098.
51 A. Schäfer, H. Horn and R. Ahlrichs, *J. Chem. Phys.*, 1992, **97**, 2571.
52 (*a*) K. Eichkorn, O. Treutler, H. Öhm, M. Häser and R. Ahlrichs, *Chem. Phys. Lett.*, 1995, **240**, 283; (*b*) K. Eichkorn, O. Treutler, H. Öhm, M. Häser and R. Ahlrichs, *Chem. Phys. Lett.*, 1995, **242**, 652, erratum.
53 F. Furche, 2005, private communication.
54 B. V. Lebedev, L. Y. Tsvetkova and K. B. Zhogova, *Thermochim. Acta*, 1997, **299**, 127.